**Лекция 7.** **Обзор современных систем шифрования. Принципы построения блочных шифров. Классификация алгоритмов шифрования. Требования к блочному алгоритму шифрования. Сеть Фейстеля и проектирование S блоков.**

Современные криптосистемы можно однозначно разделить по способу использования ключей на криптосистемы с секретным ключом (симметричные) и с открытым ключом (асимметричные). Если для зашифрования и расшифрования используется один и тот же ключ, такая криптосистема называется симметричной.

К симметричным криптосистемам относятся DES, AES, ГОСТ 28147-89 и т.д. Новым направлением в криптографии стало изобретение асимметричных криптосистем с открытым ключом, таких, как RSA, DSA или Эль-Гамаль.

В асимметричных криптосистемах для зашифрования и расшифрования применяют различные, практически не выводимые друг из друга ключи, один из которых (ключ расшифрования) делается секретным, а другой (ключ зашифрования) - открытым. Этим достигается возможность передавать секретные сообщения по незащищенному каналу без предварительной передачи секретного ключа. Именно криптография с открытым ключом разорвала порочный круг симметричных шифров, когда для организации обмена секретной информацией надо было сначала произвести распределение секретных ключей.

Детально криптосистемы с открытым ключом будут рассмотрены далее, а сейчас вернемся к симметричным криптосистемам (КС).

Важнейшей составной частью КС являются шифры или процедуры взаимообратного преобразования открытого текста M в шифртекст M':

M’=E(M),
M = D(M’),

где E - функция зашифрования и D - функция расшифрования.

Общепринятым подходом в криптографии считается такое построение шифра, при котором его секретность определяется только секретностью ключа KS(правило Керкоффа). Таким образом, шифр должен быть устойчивым к взлому, даже если потенциальному криптоаналитику известен весь алгоритм шифрования, кроме значения используемого ключа, и он располагает полным текстом перехваченной шифрограммы.

Практика показала, что чем больше известен алгоритм, чем больше людей работало с ним, тем более проверенным, а значит, и надежным он становится. Так, публично известные алгоритмы сейчас выдерживают борьбу со временем, а вот засекреченные государственные шифры обнаруживают в себе множество ошибок и недочетов, т. к. всего учесть невозможно.

Общепринятая схема построения симметричных криптосистем представляет собой циклические перестановки и подстановки битов в блоке фиксированной длины, алгоритм которых определяется секретным ключом.

 ***Рис. 1. Схема построения симметричных криптосистем***

Алгоритм шифрования считается стойким, если, имея закрытые данные, и зная секретный ключ, невозможно получить информацию об открытых данных. Строго доказана невозможность построения абсолютно стойкого шифра, за исключением случая, когда размер секретного ключа равен (или больше) размеру шифруемых данных. Этот случай трудно реализуем на практике, т.к. реально применяемые и доступные на рынке средства криптографической защиты используют шифры, для которых задача восстановления открытого текста по закрытому является трудно вычислимой, то есть требует настолько больших ресурсов, что атака становится экономически нецелесообразной.

Среди симметричных шифров наиболее известны и часто используемы следующие (размер блока в битах обозначен как b, число циклов- r, а длина ключа - l):

**DES** - государственный стандарт США (b = 64, r = 16, l = 56). В настоящее время доказана недостаточная стойкость DES против атаки методом простого перебора.
**Triple DES и DESX** (b = 64, r = 16, l = 168;112) - последовательное применение алгоритма DES с разными ключами, что обеспечивает значительную устойчивость к взлому.
**IDEA** - (b = 64, r = 8, l = 128). Активные исследования его стойкости выявили в нем ряд слабых ключей, однако вероятность их использования пренебрежимо мала.
**RC5** - параметризованный шифр с переменными размером блока (b I), количеством циклов (r 255) и числом битов ключа (l  2040). Исследования его стойкости показали, что при b = 64 он недоступен для дифференциального криптоанализа при r  12 и для линейного криптоанализа при r  7.

**ГОСТ 28147-89** - российский стандарт шифрования данных (b = 64, r = 32, l = 256). Для ГОСТа было найдено множество слабых ключей, значительно снижающих его эффективную стойкость в простых режимах шифрования. Оценка криптостойкости ГОСТа затруднена также тем фактом, что важнейшая часть алгоритма - узлы замены или S-боксы в терминологии шифра DES - не описана в стандарте и законы ее генерации остаются неизвестными. В то же время доказано, что высока вероятность получения слабых узлов замены, упрощающих криптоанализ данного шифра.
**Blowfish** - это 64-битовый блочный шифр, разработанный Шнайером (Schneier) в 1993 г., реализуется посредством перестановок и замен, зависимых от ключа. Все операции основаны на операциях XOR и прибавлениях к 32-битовым словам (XORs and additions on 32-bit words). Ключ имеет переменную длину (максимально 448 бит) и используется для генерации нескольких подключевых массивов (subkey arrays). Шифр был создан специально для 32-битовых машин и существенно быстрее DES.

Сейчас в США принят новый стандарт шифрования AES. Был проведен конкурс среди алгоритмов шифрования, в котором победил и лег в основу AES – Rijndael. Rijndael представляет собой итеративный блочный шифр, имеющий переменную длину блоков и различные длины ключей.

В мире разработано, опубликовано и исследовано достаточно большое число симметричных алгоритмов (табл. 1), из которых лишь DES и его модификация Triple DES были достаточно проверены временем. В таблицу не включены малоизвестные и слабоизученные алгоритмы, такие, как Safer, и т.д.

**Таблица 1. Обзор симметричных методов шифрования**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Метод** | **Длина ключа, бит** | **Размер блока, бит** | **Затраты на подбор ключа, MIPS x лет** | **Примечание** |
| DES | 56 | 64 | 5 102 | Разработан в 1977 г. фирмой IBM по заказу правительства США. За 20 лет не найдено способа взломать шифр, кроме полного перебора в среднем 25% всех ключей, но при современных возможностях он позволяет достичь успеха  |
| Triple DES | 128 (112) | 64 | 1018 | Трехкратное повторение алгоритма DES с разными ключами. Эффективная длина ключа 112 бит. |
| IDEA | 128 | 64 | 1021 | Разработан в 1992 г. Lai и Massey. Не взломан до настоящего времени |
| ГОСТ 28147-89 | 256 | 64 | нет данных | Является Государственным стандартом в России |
| RC5 | 40 и выше | 32 и выше | 103 и выше | 40-битовый ключ был взломан перебором в 1997 г. за 3,5 ч, 48-битовый ключ - за 313 ч |
| Blowfish | до 448 | 64 | нет данных | Разработан Шнайером (Schneier) в 1993 г. Это шифр Файстела (Feistel) был создан специально для 32-битовых машин и существенно быстрее DES |
| AES (Rijndael) | Длина ключа и длина блока могут быть 128, 192 или 256 бит, независимо друг от друга | Предложен криптоаналитиками Joan Daemen и Vincent Rijmen.Алгоритм не имеет известных слабостей в защите (по данным NIST). |

В настоящее время симметричные алгоритмы с длиной ключа более 100 бит (Triple DES и IDEA и т.д.) не являются невскрываемыми. Отечественный алгоритм ГОСТ по сравнению с ними отличается повышенной сложностью как при генерации узлов замены, так и при генерации ключей. Также для алгоритма ГОСТ существует большая вероятность генерации нестойкого ключа, что в некоторых режимах шифрования снижает его эффективную длину ключа с 2256 до 262.

Triple DES является более проверенным, чем IDEA алгоритмом и обеспечивает приемлемую скорость работы. Алгоритм Triple DES представляет собой трехкратное применение алгоритма DES к одним данным, но с разными ключами.

В Россию DES проник и достаточно широко практически используется как неотъемлемая деталь различных программных и аппаратных средств, из которых наиболее широко известны система S.W.I.F.T., секретные модули VISA и EUROPAY, секретные модули банкоматов и торговых терминалов и, наконец, смарт-карты. Особенно напряженные дискуссии вокруг алгоритмов шифрования данных вызваны именно смарт-картами. При этом есть серьезные основания считать, что надежность отечественных криптосистем конверсионного происхождения будет превосходить зарубежные аналоги.

Алгоритм ГОСТ 28147-89 построен по тому же принципу, что и DES, это классический блочный шифр с секретным ключом, однако отличается от DES большей длиной ключа, большим количеством раундов и более простой схемой построения самих раундов. В табл. 2 приведены его основные параметры, для удобства - в сравнении с параметрами DES.

**Принципы построения блочных шифров.**

**Блочные криптосистемы** разбивают текст сообщения на отдельные блоки и затем осуществляют преобразование этих блоков с использованием ключа. Используют простую замену блоков. Основные процедуры, используемые при получении таких шифров, сводятся к следующему:

* Рассеивание – изменение любого знака открытого текста или ключа влияет на большое число знаков шифротекста, что скрывает статистические свойства открытого текста
* Перемешивание – использование преобразований, затрудняющих получение статистических зависимостей между шифротектстом и открытым текстом.

В отличие от [шифроблокнота](http://dic.academic.ru/dic.nsf/ruwiki/1837381), где длина ключа равна длине сообщения, блочный шифр способен [зашифровать](http://dic.academic.ru/dic.nsf/ruwiki/6281) одним ключом одно или несколько сообщений, суммарной длиной больше, чем длина ключа. Передача малого по сравнению с сообщением ключа по зашифрованному каналу — задача значительно более простая и быстрая, чем передача самого сообщения или ключа такой же длины, что делает возможным его повседневное использование. Однако, при этом шифр перестает быть невзламываемым. От [поточных](http://dic.academic.ru/dic.nsf/ruwiki/109612) шифров работа блочного отличается обработкой бит группами, а не потоком. При этом блочные шифры надёжней, но медленее поточных. Симметричные системы обладают преимуществом над асимметричными в скорости шифрования, что позволяет им оставаться актуальными, несмотря на более слабый механизм передачи ключа (получатель должен знать секретный ключ, который необходимо передать по уже налаженному зашифрованному каналу. В то же время, в асимметричных шифрах открытый ключ, необходимый для шифрования, могут знать все, и нет необходимости в передачи ключа шифрования).

К достоинствам блочных шифров относят сходство процедур [шифрования и расшифрования](http://dic.academic.ru/dic.nsf/ruwiki/6281), которые, как правило, отличаются лишь порядком действий. Это упрощает создание устройств шифрования, так как позволяет использовать одни и те же блоки в цепях шифрования и расшифрования. Гибкость блочных шифров позволяет использовать их для построения других криптографических примитивов: [генератора псевдослучайной последовательности](http://dic.academic.ru/dic.nsf/ruwiki/6425), [поточного шифра](http://dic.academic.ru/dic.nsf/ruwiki/109612), [имитовставки](http://dic.academic.ru/dic.nsf/ruwiki/592444) и [криптографических хэшей](http://dic.academic.ru/dic.nsf/ruwiki/328468).

### Требования к блочному алгоритму шифрования

К современным алгоритмам блочного шифрования предъявляют достаточно жесткие требования, связанные с областью применения, возможностью реализации на различных вычислительных платформах и другими факторами. Рассмотрим основные из требований.

1. Алгоритм должен обеспечивать высокий уровень стойкости, и эта стойкость не должна основываться на сохранении втайне самого алгоритма.
2. Незначительное изменение исходного сообщения должно приводить к существенному изменению зашифрованного сообщения даже при использовании одного и того же ключа.
3. Алгоритм должен успешно противостоять атакам по выбранному тексту, то есть таким, чтобы нельзя было узнать ключ, даже зная достаточно много пар (зашифрованное сообщение, незашифрованное сообщение), полученных при шифровании с использованием данного ключа.
4. Алгоритм шифрования должен иметь возможность быть реализованым на различных платформах, которые предъявляют различные требования. Для наиболее быстрых приложений используется специальная аппаратура. Несмотря на это, программные реализации применяются также достаточно часто. Поэтому алгоритм должен допускать эффективную программную реализацию на универсальных микропроцессорах. Алгоритм должен также работать на микроконтроллерах и других процессорах среднего размера.
5. Алгоритм должен использовать простые операции, которые эффективны на микропроцессорах, т.е. исключающее или, сложение, табличные подстановки, умножение по модулю. Не должно использоваться сдвигов переменной длины, побитных перестановок или условных переходов.
6. Алгоритм должен эффективно реализовываться на специализированной аппаратуре, предназначенной для выполнения операций шифрования и расшифрования, то есть реализация алгоритма в виде электронных устройств должна быть экономичной.
7. Алгоритм шифрования должен быть применим во многих приложениях. Алгоритм должен быть эффективен при шифровании файлов данных или большого потока данных, при создании определенного количества случайных битов, а также должна быть возможность его использования для формирования односторонней хеш-функции1.
8. Алгоритм должен быть простым для написания кода, чтобы минимизировать вероятность программных ошибок. Также это дает возможность анализа и уменьшает закрытость алгоритма.
9. Алгоритм должен допускать любую случайную строку битов нужной длины в качестве возможного ключа (это называется *иметь плоское пространство ключей* ). Не должно быть "слабых" ключей, облегчающих криптоанализ.
10. Алгоритм должен легко модифицироваться для различных уровней безопасности и удовлетворять как минимальным, так и максимальным требованиям.

Некоторое уточнение необходимо сделать относительно пункта 1, требующего высокую криптостойкость алгоритма шифрования. Обычно под "высокой криптостойкостью" понимают, что шифр должен быть стоек по отношению к атаке по выбранному тексту. Это автоматически подразумевает его стойкость по отношению к атакам по шифротексту и по известному тексту. Однако известно, что при атаке по выбранному тексту шифр всегда может быть взломан путем перебора ключей. Поэтому требование стойкости шифра можно уточнить следующим образом: "Шифр стоек (при атаке по выбранному тексту), если для него не существует алгоритма взлома, существенно более быстрого, чем прямой перебор ключей". Интересно, что по состоянию на сегодняшний день ни для одного используемого шифра не доказано строго соответствие этому определению стойкости.

### Сеть Фейстеля

На [рис](http://www.intuit.ru/studies/courses/691/547/lecture/12375?page=2#image.3.1)унке была представлена общая структура блочного алгоритма шифрования. Понятно, что само преобразование данных выполняется в раундах или шагах шифрования. Какие же действия надо выполнить в одном раунде, чтобы в результате выполнения всего алгоритма получить надежно зашифрованные данные?

Раунд, организованный по сети Фейштеля имеет следующую структуру. Входной блок делится на несколько частей равной длины. Эти части блока называются ветвями. Так, например, если блок имеет длину 64 бита, используются две ветви по 32 бита каждая. Ветви обрабатываются по отдельности, после чего осуществляется циклический сдвиг всех ветвей влево. В случае двух ветвей каждый раунд имеет структуру, показанную на рис.



**Рис.** i-й раунд сети Фейштеля

Функция F называется *образующей*. Каждый раунд состоит из вычисления функции F для одной ветви и побитового выполнения операции "сумма по модулю 2" результата F с другой ветвью. После этого ветви меняются местами. Число раундов может быть различным для разных алгоритмов. В некоторых алгоритмах рекомендуется от 8 до 32 раундов, в других – больше. В целом увеличение количества раундов увеличивает криптостойкость алгоритма. Возможно, эта особенность и повлияла на столь активное распространение сети Фейштеля, так как для большей криптостойкости достаточно просто увеличить количество раундов, не изменяя сам алгоритм. В последнее время количество раундов не фиксируется, а лишь указываются рекомендуемые пределы.

В последнее время все чаще используются различные разновидности сети Фейстеля для 128-битного блока с четырьмя ветвями. Увеличение количества ветвей, а не размерности каждой ветви связано с тем, что наиболее популярными до сих пор остаются процессоры с 32-разрядными словами, следовательно, оперировать 32-разрядными словами эффективнее, чем с 64-разрядными.

В блочных алгоритмах, построенных на основе сети Фейстеля, основной операцией является вычисление образующей функции F. Эта функция использует подключ раунда и одну ветвь входного блока для вычисления результата. Именно тем, как определяется функция F, системы шифрования и отличаются друг от друга. В некоторых алгоритмах введены также начальные преобразования входного блока данных, придающие некоторую "случайность" входному тексту (это называется *рандомизацией* данных.) Рандомизация производится для того, чтобы уменьшить естественную избыточность входного сообщения.

Сила большинства сетей Фейстела - и особенно их устойчивость к дифференциальному и линейному криптоанализу - непосредственно связана с их S-блоками. Это явилось причиной потока исследований, что же образует хороший S-блок.

S-блок - это просто подстановка: отображение m-битовых входов на n-битовые выходы. Ранее я упоминал об одной большой таблице отображения 64-битовых входов на 64-битовые выходы, такая таблица представляла бы собой S-блок размером 64\*64 бита. S-блок с m-битовым входом и n-битовым выходом называется m\*n-битовым S-блоком. S-блоки обычно являются единственным нелинейным действием в алгоритме, именно они обеспечивают безопасность блочного шифра. В общем случае чем S-блоки больше, тем лучше.

В DES восемь различных 6\*4-битовых S-блоков. В Khufu и Khafre единственный 8\*32-битовый S-блок, в LOKI 12\*8-битовый S-блок, а в Blowfish и CAST 8\*32-битовые S-блоки. В IDEA S-блоком по сути является умножение по модулю, это 16\*16-битовый S-блок. Чем больше S-блок, тем труднее обнаружить статистические отклонения, нужные для вскрытия с использованием либо дифференциального, либо линейного криптоанализа. Кроме того, хотя случайные S-блоки обычно не оптимальны с точки зрения устойчивости к дифференциальному и линейному криптоанализу, сильные S-блоки легче найти среди S-блоков большего размера. Большинство случайных S-блоков нелинейны, невырождены и обладают сильной устойчивостью к линейному криптоанализу - и с уменьшением числа входных битов эта доля снижается медленно.

Размер m важнее размера n. Увеличение размера n снижает эффективность дифференциального криптоанализа, но значительно повышает эффективность дифференциального криптоанализа. Действительно, если n?2m-m, то наверняка существует линейная зависимость для входных и выходных битов S-блока. И если n?2m, то линейная зависимость существует только для выходных битов.

Заметной частью работы по проектированию S-блоков является изучение логических функций. Для обеспечения безопасности булевы функции, используемые в S-блоках, должны отвечать определенным условиям. Они не должны быть ни линейными, ни аффинными, ни даже быть близкими к линейным или аффинным. Количество нулей и единиц должно быть сбалансированным, и не должно быть никаких корреляций между различными комбинациями битов. При изменении на противоположный любого входного бита выходные биты должны вести себя независимо. Эти критерии проектирования также связаны с изучением функций изгиба: функций, которые, как может быть показано, являются оптимально нелинейными. Хотя они определены просто и естественно, их изучение очень нелегко.

Очень важным свойством представляется лавинный эффект: сколько выходных битов S-блока изменяется при изменении некоторого подмножества выходных битов. Нетрудно задать для булевых функций условия, выполнение которых обеспечивает определенный лавинный эффект, но проектирование таких функций является более сложной задачей. Строгий лавинный критерий (strict avalanche criteria, SAC) обеспечивает, что с изменением одного входного бита изменяется ровно половина выходных битов.

Несколько лет назад криптографы предложили выбирать S-блоки так, чтобы таблица распределения различий для каждого S-блока была однородной. Это обеспечило бы устойчивость к дифференциальному криптоанализу за счет сглаживания дифференциалов на любом отдельном этапе. Примером такого проектирования является LOKI. Однако такой подход иногда способствует дифференциальному криптоанализу. Действительно, лучшим подходом является минимизирование максимального дифференциала. Кванджо Ким (Kwangjo Kim) выдвинул пять критериев проектирования S-блоков, похожих на критерии проектирования S-блоков DES.

**Проектирование S блоков** .

Выбор хороших S-блоков - не простая задача, существует множество различных идей, как лучше сделать это. Можно выделить четыре главных подхода.

1. Случайно выбрать. Ясно, что небольшие случайные S-блоки небезопасны, но большие случайные S-блоки могут оказаться достаточно хороши. Случайные S-блоки с восемью и более входами достаточно сильны. Еще лучше 12-битовые S-блоки. Устойчивость S-блоков возрастает, если они одновременно являются и случайными, и зависящими от ключа. В IDEA используются большие зависящие от ключа S-блоки.
2. Выбрать и проверить. В некоторых шифрах свойства S-блоков, генерированных случайным образом, проверяются.
3. Разработать вручную. При этом математический аппарат используется крайне незначительно: S-блоки создаются с использованием интуитивных приемов. Барт Пренел (Bart Preneel) заявил, что "... теоретически интересные критерии недостаточны [для выбора булевых функций S-блоков] ...", и что "... необходимы специальные критерии проектирования" .
4. Разработать математически. S-блоки создаются в соответствии с математическими законами, поэтому они обладают гарантированной надежностью по отношению к дифференциальному и линейному криптоанализу, а также хорошими диффузными свойствами.

Существует ряд призывов объединить "математический" и "ручной" подходы, но реально, по видимому, конкурируют случайно выбранные S-блоки и S-блоки с определенными свойствами. Конечно преимуществом последнего подхода является оптимизация против известных методов вскрытия - дифференциального и линейного криптоанализа - но обеспечиваемая этим подходом степень защиты от неизвестных методов вскрытия также неизвестна. Разработчикам DES было известно о дифференциальном криптоанализе, и его S-блоки были оптимизированы соответствующим образом. Скорее всего, о линейном криптоанализе они не знали, и S-блоки DES очень слабы по отношению к такому способу вскрытия. Случайно выбранные S-блоки в DES были бы слабее против дифференциального криптоанализа, но сильнее против линейного криптоанализа.

С другой стороны случайные S-блоки могут не быть оптимальными по отношению к данным способам вскрытия, но они могут быть достаточно большими и, следовательно, достаточно надежными. Кроме того, они, скорее всего, будут достаточно устойчивы и против неизвестных способов вскрытия. Спор все еще кипит, но лично мне кажется, что S-блоки должны быть такими большими, насколько это возможно, случайными и зависеть от ключа.